

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian dimulai dengan melakukan kajian penelitian sebelumnya untuk memahami konsep yang akan digunakan dalam penelitian. Untuk membandingkan dan mendapatkan inspirasi untuk ide-ide baru, referensi yang digunakan adalah jurnal yang paling baru, terkini, dan relevan dengan penelitian yang sedang dilakukan. Penelitian sebelumnya dan penelitian baru juga dibandingkan.

Pertama, penelitian berjudul “Perancangan Aplikasi *Point Of Sales* (POS) Berbasis Android (Studi Kasus: Warkop Vape Salatiga)” yang dilakukan oleh Restiawan Yoga Pratama, Ramos Somya. Penelitian ini bertujuan untuk menciptakan sistem *point of sales* berbasis android untuk Warkop Vape salatiga. Contoh hasil yang bisa diberikan adalah ketika melakukan transaksi, proses ini bisa berlangsung dengan cepat, mudah dan akurat. Tidak ada masalah yang ditemukan selama uji coba penggunaan aplikasi[11].

Kedua, penelitian yang berjudul “Pengembangan Aplikasi Mobile Pencatatan Transaksi Penjualan Olahan Mangga Terintegrasi Berbasis Android “yang dilakukan oleh Adi Suheryadi, Kurnia Adi Cahyanto, M Lukman Sifa, Diyanto, Ade Diana P. Penelitian ini bertujuan untuk UMKM Kenanga Mandiri menggunakan aplikasi pencatatan transaksi berbasis *client-server*. Sistem ini memudahkan proses transaksi, Dengan menggunakan teknologi *barcode* dan *qr-code* yang tersedia di library Zxing[12].

Ketiga, penelitian yang berjudul Rancang bangun aplikasi *point of sale* tiket bus pada PT primajasa Perdanarayautama yang dilakukan oleh Euis Siti Nur Aisyah, Abdul Hayat, Rivka Farizi, Ahmad Ajie Sajidin. Penelitian ini bertujuan untuk merancang dan membangun aplikasi Point of Sale (POS) tiket bus pada PT Primajasa Perdanarayautama Aplikasi ini memenuhi kebutuhan pengguna, memudahkan proses transaksi penjualan tiket bus, Meningkatkan

efisiensi dan produktivitas kerja, Menyediakan laporan penjualan yang akurat dan terkini, Penelitian ini berbasis web (GUI) dan tidak menggunakan metode blackbox testing[4].

Keempat, penelitian yang berjudul Rancang Bangun Sistem Informasi *Point of Sales* Berbasis *Website* Pada Distributor Kain Hoggy Djaya yang dilakukan oleh Chandra K. Penelitian ini bertujuan untuk memenuhi kebutuhan klien, fitur khusus di luar permintaan klien telah ditambahkan untuk mendukung kinerja aplikasi. Dalam penelitian ini menggunakan *User Acceptance Test* sebagai hasil *test*[13].

Kelima, penelitian yang berjudul “Rancang Bangun Sistem Informasi *POS* (*Point of Sale*) untuk kasir menggunakan konsep Bahasa yang dilakukan oleh Samsir, Muhammad Siddik. Penelitian ini bertujuan untuk melakukan Proses penjumlahan dan pengurangan dapat dilakukan lebih cepat, tepat dan akurat sehingga terhindar dari resiko kesalahan dalam perhitungan keuangan serta pendataan stok barang, Sistem ini juga dapat melakukan proses pencarian data, penyimpanan data, pengelompokan data sehingga dalam pembuatan laporan dapat berberjalan dengan efektif dan efisien. Penelitian ini menggunakan metode *Waterfall*[14].

Keenam, penelitian yang berjudul “Rancang Bangun Aplikasi *Point Of Seles* (POS) Untuk Toko Kecil’ yang dilakukan oleh Audy A. Wansaga, Hans F. Wowor, Alwin M.Sambul. Penelitian mendapatkan hasil Aplikasi POS *platform* android yang dibangun bertujuan untuk mempermudah pemilik Toko Kecil dalam transaksi penjualan, sekaligus meminimalkan biaya pembelian perangkat pendukung seperti komputer dan alat pemindai *barcode*.

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No	Judul	Penulis	Kesamaan	Perbedaan	Kesimpulan
1	Perancangan Aplikasi <i>Point Of Sales</i> (POS) Berbasis Android (Studi Kasus: Warkop Vape Salatiga[11])	Restiawan Yoga Pratama, Ramos Somya	Perancangan sistem ini menggunakan teknologi dari firebase yaitu fitur realtime database, Aplikasi <i>Point Of Sales</i> berbasis Android adalah hasil dari penelitian ini.	Penelitian ini memanfaatkan teknologi barcode dan qr-code yang tersedia di library Zxing, sedangkan yang peneliti lakukan tidak menggunakan barcode	Pengujian sistem yang dilakukan oleh peneliti sebelumnya dapat disimpulkan bahwa aplikasi sudah sesuai dengan kebutuhan. Wawancara dengan pemilik warkop menunjukkan bahwa aplikasi sudah sesuai dengan kebutuhan, seperti saat melakukan proses transaksi.

2	Pengembangan Aplikasi Mobile Pencatatan Transaksi Penjualan Olahan Mangga Terintegrasi Berbasis Android[12]	Adi Suheryadi, Kurnia Adi Cahyanto, M Lukman Sifa, Diyanto, Ade Diana P.	Sistem yang dibangun memiliki kesamaan yaitu berbasis Android.	Database yang digunakan pada penelitian sebelumnya menggunakan MySQL, tetapi peneliti ini menggunakan Firebase, yang tidak menggunakan SQL.	UMKM Kenanga Mandiri menggunakan aplikasi pencatatan transaksi berbasis client-server. Sistem ini memudahkan proses transaksi.
3	Rancang bangun aplikasi <i>point of sales</i> tiket bus pada pt primajasa perdanarayutama[4]	Euis Siti Nur Aisyah, Abdul Hayat, Rivka Farizi, Ahmad Ajie Sajidin	Penelitian ini memiliki kesamaan tujuan, Bertujuan agar dapat mempercepat proses pelayanan transaksi penjualan ,menghasilkan laporan	Penelitian ini berbasis web (GUI) dan tidak menggunakan metode <i>blackbox testing</i> .	Aplikasi POS MODA dapat membantu perusahaan transportasi yang tidak dapat menawarkan pelanggannya

			keuangan secara langsung.		pemesanan tiket secara <i>online</i> ..
4	Rancang Bangun Sistem Informasi <i>Point of Sales</i> Berbasis Website Pada Distributor Kain Hoggy Djaya[13]	Chandra K	Penelitian sebelumnya menunjukkan bahwa sistem Point of Sale dapat membantu menyelesaikan masalah saat ini.	Penelitian ini menggunakan <i>User Acceptance Test</i> sebagai hasil testing.	Calon pengguna telah menyetujui sistem Point of Sale yang dibuat melalui internet. Untuk memenuhi kebutuhan klien, fitur khusus di luar permintaan klien telah ditambahkan untuk mendukung kinerja aplikasi.
5	Rancang bangun sistem informasi pos ( <i>point of sales</i> ) untuk kasir menggunakan konsep bahasa	Samsir, Muhammad Siddik	Penelitian ini bertujuan untuk membangun sistem informasi POS (Point of	Penelitian ini menggunakan metode yang berbeda dari penulis, yaitu	Konsep OOP dalam sistem aplikasi POS untuk kasir meningkatkan kinerja layanan karena

	pemrograman orientasi objek[14]		Sale) yang dapat membantu proses transaksi penjualan di sebuah usaha.	menggunakan metode waterfall	seluruh proses pencatatan transaksi penjualan, dan pengolahan data
6	Rancang Bangun Aplikasi <i>Point Of Sale</i> (POS) Untuk Toko Kecil[15]	Audy A. Wansaga, Hans F. Wowor, Alwin M.Sambul	Penelitian ini memiliki kesamaan menggunakan firebase sebagai databasenya	Studi kasus yang diteliti memiliki perbedaan, dimana penelitian sebelumnya untuk toko kecil, penelitian yang akan dilakukan untuk Kedai Marjok	Aplikasi POS berplatform android yang dibangun bertujuan untuk mempermudah pemilik Toko Kecil dalam transaksi penjualan, sekaligus meminimalkan biaya pembelian perangkat pendukung seperti komputer dan alat pemindai barcode

Dari tabel di atas dapat disimpulkan Aplikasi POS berbasis Android yang dibangun menggunakan metode *Rapid Application Development* dapat memenuhi kebutuhan pengguna untuk mengelola transaksi penjualan, mengelola stok barang, dan menghasilkan laporan penjualan. Aplikasi ini dapat menjadi solusi yang tepat bagi bisnis yang ingin meningkatkan efisiensi dan produktivitasnya.

## 2.2 Landasan Teori

### 2.2.1 Rancang Bangun

*Rancang bangun* adalah proses merancang atau membuat ide dan model untuk produk atau sistem baru dengan mempertimbangkan berbagai hal seperti fungsi, keamanan, efisiensi, dan estetika. Ini diperlukan untuk membuat produk atau sistem yang berfungsi dengan baik dan mudah digunakan oleh pengguna[16]. Rancang bangun merupakan suatu aktivitas yang membutuhkan kreativitas, inovasi, dan pengetahuan yang luas. Untuk mencapai hasil yang optimal, rancang bangun harus dilakukan dengan sistematis dan mendetail. Selain itu, rancang bangun harus sesuai dengan standar dan pedoman yang berlaku agar sesuai dengan kebutuhan dan harapan pengguna[17].

### 2.2.2 Point of Sale

*Point of Sale* adalah lokasi tempat konsumen membayar untuk produk atau layanan yang mereka beli dan tempat mereka mungkin harus membayar pajak atas penjualan. POS dapat berlangsung secara fisik atau digital, dengan bukti pembayaran yang dicetak atau dikirim secara elektronik[18]. Sebelum pengembangan sistem *Point of Sale*, sebagian besar orang menggunakan *cash register* dan bahkan melakukan transaksi secara manual. Fungsi *cash register* sendiri menjadi kurang memadai, jadi sekarang POS dibuat[19]. POS membantu analisis dan pengambilan keputusan dan meningkatkan produktivitas, akurasi, dan keamanan bisnis.[20]

### 2.2.3 Rapid Application Development

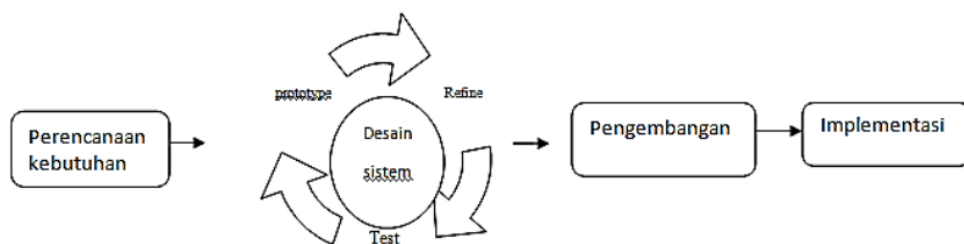
*RAD* merupakan metode pengembangan perangkat lunak yang mengutamakan kecepatan dan iterasi, dimana model kerja dibangun sejak awal untuk menyesuaikan dengan kebutuhan pengguna dan kemudian dibuat[21]. *RAD* merupakan adaptasi cepat dari model *waterfall*, dengan menggunakan pendekatan pembuatan komponen. *RAD* adalah kombinasi dari berbagai teknik *prototyping* dan metode pengembangan *joint application* untuk mempercepat



pembuatan sebuah sistem[22]. berikut ini adalah tahapan pengembangan aplikasi dari tiap fase RAD.

1. Perancangan kebutuhan: Tahapan ini merupakan tahap awal dalam suatu pengembangan sistem, dimana pada tahap ini dilakukan identifikasi masalah dan pengumpulan data yang diperoleh dari pengguna atau *stakeholder* pengguna yang bertujuan untuk mengidentifikasi maksud akhir atau tujuan dari sistem dan kebutuhan informasi yang diinginkan. Pada tahap ini keterlibatan kedua belah sangatlah penting dalam mengidentifikasi kebutuhan untuk pengembangan suatu sistem.
2. Desain Sistem: Di dalam tahap desain sistem, keaktifan pengguna yang terlibat sangatlah penting untuk mencapai tujuan karena pada tahapan ini dilakukan proses desain dan proses perbaikan desain secara berulang-ulang apabila masih terdapat ketidaksesuaian desain terhadap kebutuhan pengguna yang telah diidentifikasi pada tahapan sebelumnya. Luaran dari tahapan ini adalah spesifikasi software yang meliputi organisasi di dalam sistem secara umum, struktur data, dan lain-lain.
3. Pengembangan: Pada tahap ini desain sistem yang telah dibuat dan disepakati, diubah ke dalam bentuk aplikasi versi beta sampai dengan versi final. Pada tahapan ini juga programmer harus terus-menerus melakukan kegiatan pengembangan dan integrasi dengan bagian-bagian lainnya sambil terus mempertimbangkan feedback dari pengguna atau klien. Jika proses berjalan lancar maka dapat berlanjut ke tahapan berikutnya, sedangkan jika aplikasi yang dikembangkan belum menjawab kebutuhan, programmer akan kembali ke tahapan desain sistem.

4. Implementasi: Tahapan ini merupakan tahapan dimana programmer menerapkan desain dari suatu sistem yang telah disetujui pada tahapan sebelumnya. Sebelum sistem diterapkan, terlebih dahulu dilakukan proses pengujian terhadap program untuk mendeteksi kesalahan yang ada pada sistem yang dikembangkan. Pada tahap ini biasa memberikan tanggapan akan sistem yang sudah dibuat dan mendapat persetujuan mengenai sistem tersebut[15].



Gambar 2.1 Tahapan RAD [9]

#### 2.2.4 Figma

Figma merupakan alat desain yang sering dipakai untuk menciptakan antarmuka aplikasi mobile, desktop, website dan sebagainya. Figma dapat dijalankan di sistem operasi windows, linux atau mac dengan koneksi internet.

Selain memiliki fitur-fitur lengkap seperti *Adobe XD*, Figma juga memiliki kelebihan yaitu dapat memungkinkan pekerjaan yang sama dilakukan oleh beberapa orang secara bersamaan meskipun berada di tempat yang berlainan. Hal ini dapat disebut sebagai kerja tim dan karena kemampuan aplikasi figma inilah yang membuat aplikasi ini menjadi favorit banyak *UI/UX* designer untuk membuat *prototype* website atau aplikasi dengan cepat dan efisien.

#### 2.2.5 Android

*Android* adalah sistem operasi berbasis Linux yang dibuat untuk ponsel pintar dan tablet dengan layar sentuh[21]. Karena bersifat sumber terbuka, kode program Android dapat dilihat, diubah, dan disebarluaskan oleh siapapun tanpa perlu membayar lisensi. Hal ini memberikan kesempatan bagi para

pengembang dan produsen untuk menciptakan variasi dan inovasi pada sistem operasi sesuai dengan kebutuhan dan selera mereka[23].

#### **2.2.5.1 JavaScript**

*JavaScript* (JS) merupakan bahasa pemrograman tingkat tinggi dan dinamis yang sangat penting dalam pengembangan web. Dikembangkan oleh Brendan Eich pada tahun 1995, *JavaScript* awalnya dirancang untuk dijalankan di sisi klien (*client-side*) pada *browser* web. Namun, dengan kemajuan teknologi, *JavaScript* kini juga dapat digunakan di sisi server (*server-side*) melalui platform seperti *Bun.js*. *JavaScript* memainkan peran vital dalam menciptakan halaman *web* yang interaktif dan dinamis. Java yang telah ada sebelumnya adalah dua keunggulan utamanya.

*JavaScript* pertama kali diperkenalkan oleh Netscape Communications sebagai LiveScript, namun kemudian diubah namanya menjadi JavaScript untuk tujuan pemasaran. Bahasa ini dengan cepat diadopsi oleh komunitas pengembang web dan diresmikan melalui spesifikasi ECMAScript oleh [24].

#### **2.2.6 UML**





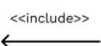

UML (Unified Modeling Language) adalah bahasa pemodelan visual yang digunakan secara luas yang dibuat dengan tujuan menawarkan cara yang konsisten untuk menunjukkan desain sistem. Berbagai macam diagram tersedia dalam UML, yang termasuk dalam tiga kategori utama: diagram perilaku, diagram interaksi, dan diagram struktur[25]. UML merupakan salah satu standar bahasa yang paling populer di industri untuk analisis, desain, dan definisi persyaratan pemrograman berorientasi objek[26]. UML merupakan bahasa pemodelan yang digunakan untuk menentukan, mendokumentasikan, dan mengembangkan perangkat lunak. Standar penyusunan sistem UML mencakup proses bisnis, skema basis data, kelas bahasa pemrograman tertentu, dan komponen sistem perangkat lunak yang diperlukan. Diagram UML antara lain [27].

### 1. Use Case Diagram

*Use case* merupakan teknik pemodelan visual yang digunakan untuk menentukan, mencatat, dan mengembangkan sistem. *Use case* menggambarkan bagaimana satu atau lebih peran (actor) berinteraksi dengan sistem untuk mencapai tujuan tertentu[28]. Diagram use case berguna untuk mengelola implementasi (persyaratan) yang sudah ada, karena *use case* baru selalu menciptakan implementasi baru seiring dengan semakin jelasnya sistem yang dianalisis dan dirancang. Selain itu, penggunaan simbol dan simbol dalam diagram memfasilitasi komunikasi antara pengembang dan pelanggan[29].

Dalam *use case*, tampilan eksternal *view* sistem yang akan dibuat dalam model dapat digambarkan. Karena lebih sempit dari model, diagram tidak sama dengan model[30]. Tabel 2.2 merupakan *use case* yang menggunakan symbol diagram :








Tabel 2.2 Diagram Use Case[29].

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan use case
	Use case : Abstraksi dan interaksi antara sistem dan aktor
	Association : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

## 2. Class Diagram

*Class diagram* kelas adalah jenis diagram Unified Modeling Language (UML) yang digunakan untuk memodelkan struktur statis sistem berorientasi objek. *Class Diagram* menggambarkan kelas, properti, operasi, dan hubungan antar objek dalam sistem [31]. Diagram kelas adalah representasi struktur kelas yang terdapat dalam sistem. Tabel 2.3 merupakan simbol yang digunakan pada *class diagram*.






Tabel 2.3 Simbol *Class Diagram*[29]

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

## 3. Activity Diagram

*Activity diagram* merupakan diagram UML yang digunakan untuk memodelkan perilaku dinamis dari sebuah system. Activity Diagram merupakan aktivitas, objek, transisi keadaan, dan peristiwa. Dengan kata lain, aktivitas dalam diagram alur kerja menggambarkan perilaku sistem sehubungan dengan aktivitas tersebut[29]. Tabel 2.4 merupakan simbol yang digunakan pada *activity diagram*.







Tabel 2.4 *Activity Diagram*[29]

Nama	Simbol	Fungsi
Initial State		Menggambarkan awal dimulainya suatu aliran aktivitas
Final State		Menggambarkan berakhirnya suatu aliran aktivitas
Activity		Menggambarkan aktivitas yang dilakukan dalam suatu aliran aktivitas
Decision		Menggambarkan pilihan kondisi atau cabang-cabang aktivitas tertentu
Transition		Berguna untuk menghubungkan satu komponen dengan komponen lainnya.

#### 4. *Sequence Diagram*

*Sequence Diagram* merupakan diagram yang digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem. *Sequence diagram* merupakan ilustrasi langkah-langkah, termasuk urutan perubahan secara logis yang harus dilakukan untuk mencapai sesuatu sesuai dengan use case diagram[32]. Tabel 2.5 merupakan simbol yang digunakan pada *sequence diagram*.

Tabel 2.5 *Sequence Diagram*[29]

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control &amp; A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

#### 2.2.7 *Expo*

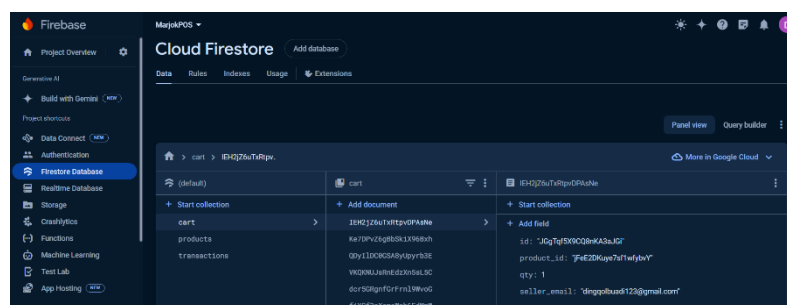
Expo merupakan platform pengembangan aplikasi mobile yang menyediakan lingkungan pengembangan berbasis JavaScript dan React Native. Expo memfasilitasi akses pengembang terhadap beragam alat pengembangan, termasuk fitur-fitur tambahan seperti *live reloading* dan debugging tools yang meningkatkan efisiensi dalam proses pengembangan

aplikasi[33]. Selain itu, Expo mendukung pengembangan aplikasi lintas *platform* tanpa perlunya penulisan ulang kode secara terpisah, memungkinkan pengembang untuk fokus pada pengembangan aplikasi dengan cepat dan efisien. Komunitas pengembang yang aktif dan dokumentasi yang lengkap juga menjadi keunggulan Expo, memberikan akses mudah terhadap bantuan, pengetahuan, dan pemahaman yang mendalam dalam pengembangan aplikasi mobile. Dengan Expo, pengembang dapat memanfaatkan kelebihan React Native tanpa konfigurasi yang rumit, mempercepat proses pengembangan aplikasi, dan meningkatkan kualitas aplikasi yang dihasilkan[34].

## 2.2.8 Firebase Firestore Database

Firebase Firestore *Database* merupakan layanan basis data NoSQL yang dibuat oleh Google sebagai bagian dari *platform Firebase*. Firestore dirancang untuk menyimpan, menyinkronkan, dan mengelola data untuk aplikasi web dan seluler dalam skala besar. Basis data ini menyediakan fitur pembaruan real-time, skalabilitas tinggi, serta kemudahan integrasi dengan berbagai layanan Firebase lainnya[35].

Adapun Fitur-fitur Firebase Realtime Database tertera pada gambar 2.2.



Gambar 2.2 Fitur *Firebase* [35]

## 2.2.9 User Persona

User persona merupakan representasi fiktif dari tipe pengguna yang ideal dalam sebuah produk atau layanan. Persona ini diciptakan berdasarkan data riset pengguna yang nyata, dan digunakan untuk memahami kebutuhan, pengalaman, perilaku, dan tujuan dari pengguna. Dengan memahami persona

pengguna, tim pengembang dapat membuat keputusan desain dan pengembangan yang lebih baik dan lebih tepat sasaran[36].

User persona adalah alat yang sangat berharga dalam desain dan pengembangan produk yang berpusat pada pengguna. Dengan memahami siapa pengguna, apa yang mereka butuhkan, dan bagaimana mereka berinteraksi dengan produk, tim pengembang dapat menciptakan produk yang lebih efektif, efisien, dan memuaskan[37].

### 2.2.10 System Usability Scale (SUS)

Metode *System Usability Scale* (SUS) adalah sebuah teknik penilaian yang digunakan untuk memberikan sudut pandang subjektif tentang sejauh mana sistem dianggap berguna oleh pengguna. Skor yang diperoleh dari metode SUS dapat digunakan sebagai salah satu faktor dalam menilai apakah sebuah aplikasi pantas digunakan. SUS terdiri dari sebuah kuesioner yang terdiri dari 10 pernyataan yang digunakan untuk mengevaluasi suatu produk. Kuesioner SUS mencakup pernyataan positif dan negatif[38]. Pernyataan dengan nomor ganjil berisi pernyataan positif, sementara pernyataan dengan nomor genap berisi pernyataan negatif. Ketika responden mengisi kuesioner, yang biasanya disebarluaskan melalui platform seperti Google Form, mereka diminta untuk memberikan jawaban berdasarkan skala Likert yang mencakup pilihan dari "sangat setuju", "setuju", "netral", "tidak setuju", hingga "sangat tidak setuju".

	STS	TS	RG	ST	SS
1. Saya berpikir akan menggunakan sistem ini lagi.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5

Gambar 2.3 Contoh Pertanyaan dan pilihan jawaban SUS

Tata cara penghitungan skor untuk berlaku pada 1 responden. Untuk perhitungan selanjutnya, skor SUS dari masing-masing responden dicari skor rata-ratanya dengan menambahkan semua skor dan dibagi dengan jumlah responden. Berikut rumus menghitung skor sus:



$$\text{Skor SUS} = 2.5 * (\sum x / n)$$

Dalam rumus di atas:

$\bar{x}$  = Skor rata rata

$\sum x$  = Jumlah semua skor individu dari responden.

$n$  = Jumlah responden.

### 2.2.11 Black Box Testing

*Black Box Testing* adalah cara untuk menguji perangkat lunak yang hanya memeriksa fungsi aplikasi tanpa mengetahui struktur atau mekanisme internalnya. Pengujian ini dilakukan dengan memberikan input dan mengamati output yang dihasilkan oleh sistem yang diuji. Tujuan dari black box testing adalah untuk memastikan apakah sistem berjalan sesuai dengan spesifikasi atau kebutuhan yang diberikan[39].

*Black Box Testing* digunakan untuk menunjukkan kesalahan sistem, seperti kesalahan dalam fungsinya. Pengujian dilakukan dengan masukan data acak untuk mendapatkan hasil yang pasti. Dalam hal ini, pasti berarti bahwa sistem akan melakukan penolakan ketika terjadi kesalahan, sedangkan ketika data yang dimasukkan benar dapat diterima atau dimasukkan ke database[40].

#### a. Cara Kerja *Black Box Testing*

Dengan *black box testing*, sistem diuji sesuai dengan kebutuhan yang ada dan ditantang dengan berbagai input yang tidak valid, agar pengguna dapat memberikan data yang benar. Tujuan pengujian adalah untuk menemukan kesalahan seperti: fungsi yang salah, antarmuka yang buruk, struktur data atau akses basis data eksternal yang bermasalah, dan kinerja yang rendah[41].

#### b. Teknik *Black Box Testing*

##### 1. *Decision Table*

*Decision Table* adalah sebuah cara pengujian yang menggunakan tabel yang berisi matriks dari setiap kasus uji yang dibentuk. Setiap kolomnya menunjukkan sebuah kombinasi unik dari kondisi dan

tindakan yang dihasilkan dari kasus uji itu. Tabel ini dibuat untuk mengetahui struktur logika dari sebuah program.

## 2. *All-Pairs Testing*

*All-Pairs Testing* adalah sebuah cara pengujian kombinatorial yang menguji semua kemungkinan kombinasi diskrit dari setiap pasangan parameter yang masuk ke sistem atau algoritma yang ada pada perangkat lunak.

## 3. *State Transition Table*

*State Transition Table* adalah sebuah cara pengujian yang menggunakan model sistem yang terdiri dari status, transisi, kejadian, dan aksi. Model ini biasanya direpresentasikan dalam bentuk diagram transisi status. Kasus uji didesain untuk memeriksa validitas transisi antara status. Kasus uji tambahan juga didesain untuk menguji transisi yang tidak termasuk atau tidak dispesifikasikan.

## 4. *Equivalence Partitioning*

*Equivalence Partitioning* adalah sebuah cara pengujian yang membagi input data menjadi beberapa kelompok yang setara berdasarkan persyaratan atau spesifikasi fungsional. Setiap kelompok dianggap memiliki perilaku yang sama, sehingga hanya perlu diuji dengan satu nilai perwakilan. Cara ini bertujuan untuk mengurangi jumlah kasus uji yang diperlukan tanpa mengurangi efektivitas pengujian.

## 5. *Boundary Value Analysis*

*Boundary Value Analysis* adalah sebuah cara pengujian yang fokus pada nilai-nilai batas pada rentang input data. Nilai-nilai batas ini meliputi nilai minimum, maksimum, tepat di atas atau di bawah batas, dan nilai-nilai tidak valid. Cara ini didasarkan pada asumsi bahwa kesalahan sering terjadi pada batas-batas rentang input data[29]